

Docket No. AUS920010996US1

CLAIMS:

What is claimed is:

1. A method in a data processing system for generating
5 a generic compilation interface from a first
object-oriented software package, said method comprising
the steps of:
 identifying all public classes included in said
first software package;
10 for each of said public classes, identifying all
public entities included in each of said public classes;
 removing all references to software that is defined
in a second software package from said public entities
included in each of said public classes;
15 generating an equivalent public class for each of
said identified public classes, said equivalent public
class including equivalent public entities that include
no references to said software defined in said second
package;
20 compiling each of said equivalent public classes;
and
 generating a compilation interface for said first
package including each of said compiled equivalent public
classes.
25
2. The method according to claim 1, wherein said step
of identifying all public entities included in each of
said public classes further comprises the step of
identifying all entities included in each of said public
30 classes that include a public modifier.

Docket No. AUS920010996US1

3. The method according to claim 1, further comprising the steps of:

determining whether each of said entities includes a native attribute;

5 in response to a determination that each of said entities includes a native attribute, removing said native attribute from each of said entities; and

generating equivalent entities that include no native attributes.

10

4. The method according to claim 1, wherein the step of identifying all public entities included in each of said public classes further comprises the step of identifying all public methods included in each of said public

15

classes.

5. The method according to claim 1, wherein the step of identifying all public entities included in each of said public classes further comprises the step of identifying

20

all public parameters included in each of said public classes.

6. The method according to claim 1, wherein the step of identifying all public entities included in each of said

25

public classes further comprises the step of identifying all public fields included in each of said public classes.

7. The method according to claim 1, wherein said step of identifying all public classes included in said first package further comprises the step of identifying all public classes included in a Java Archive file.

30

Docket No. AUS920010996US1

8. The method according to claim 1, wherein the step of identifying all public classes included in said first package further comprises the step of identifying all public classes included in said first package utilizing a java.util.jar utility.

9. The method according to claim 1, wherein the step of for each of said public classes, identifying all public entities included in each of said public classes further comprises the step of for each of said public classes, identifying all public entities included in each of said public classes utilizing Java Reflection.

10. The method according to claim 1, wherein said step of generating an equivalent public class for each of said identified public classes further comprises the step of generating a separate .java file for each of said identified public classes.

11. The method according to claim 10, wherein said step of compiling each of said equivalent public classes further comprises the step of compiling each said .java file.

12. The method according to claim 11, wherein said step of generating a compilation interface for said first package including each of said compiled equivalent public classes further comprises the steps of:

generating a compilation Java Archive file; and
storing each said compiled .java file in said compilation Java Archive file.

Docket No. AUS920010996US1

13. The method according to claim 1, further comprising the step of utilizing said compilation interface within an Integrated Development Environment.

5 14. A data processing system for generating a generic compilation interface from a first object-oriented software package, said system comprising:

means for identifying all public classes included in said first package;

10 means for each of said public classes, for identifying all public entities included in each of said public classes;

means for removing all references to software defined in a second software package from said public entities included in each of said public classes;

15 means for generating an equivalent public class for each of said identified public classes, said equivalent public class including equivalent public entities that include no references to said software defined in said second package;

20 means for compiling each of said equivalent public classes; and

means for generating a compilation interface for said first software package including each of said compiled equivalent public classes.

25 15. The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all entities included in each of said public classes that include a public modifier.

Docket No. AUS920010996US1

16. The system according to claim 14, further comprising:

means for determining whether each of said entities includes a native attribute;

5 in response to a determination that each of said entities includes a native attribute, means for removing said native attribute from each of said entities; and

means for generating equivalent entities that include no native attributes.

10

17. The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all public methods included in each of said public classes.

15

18. The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all public parameters included in each of said public classes.

20

19. The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all public fields included in each of said public classes.

25

20. The system according to claim 14, wherein said means for identifying all public classes included in said first package further comprises means for identifying all public classes included in a Java Archive file.

30

2020010996US1

Docket No. AUS920010996US1

21. The system according to claim 14, wherein said means
for identifying all public classes included in said first
package further comprises means for identifying all
public classes included in said first package utilizing a
5 java.util.jar utility.

22. The system according to claim 14, wherein said means
for each of said public classes, for identifying all
public entities included in each of said public classes
10 further comprises means for each of said public classes,
for identifying all public entities included in each of
said public classes utilizing Java Reflection.

23. The system according to claim 14, wherein said means
15 for generating an equivalent public class for each of
said identified public classes further comprises means
for generating a separate .java file for each of said
identified public classes.

24. The system according to claim 23, wherein said means
20 for compiling each of said equivalent public classes
further comprises means for compiling each said .java
file.

25. The system according to claim 24, wherein said means
for generating a compilation interface for said first
package including each of said compiled equivalent public
classes further comprises:

means for generating a compilation Java Archive
30 file; and

means for storing each said compiled .java file in
said compilation Java Archive file.

Docket No. AUS920010996US1

26. The system according to claim 1, further comprising means for utilizing said compilation interface within an Integrated Development Environment.

5 27. A computer program product in a data processing system for generating a generic compilation interface from a first object-oriented software package, said computer program product comprising:

10 instruction means for identifying all public classes included in said first package;

instruction means for each of said public classes, for identifying all public entities included in each of said public classes;

15 instruction means for removing all references to software defined in a second software package from said public entities included in each of said public classes;

20 instruction means for generating an equivalent public class for each of said identified public classes, said equivalent public class including equivalent public entities that include no references to said software defined in said second package;

instruction means for compiling each of said equivalent public classes; and

25 instruction means for generating a compilation interface for said first package including each of said compiled equivalent public classes.

28. The product according to claim 27, wherein said instruction means for identifying all public entities
30 included in each of said public classes further comprises

Docket No. AUS920010996US1

instruction means for identifying all entities included in each of said public classes that include a public modifier.

- 5 29. The product according to claim 27, further comprising:

instruction means for determining whether each of said entities includes a native attribute;

- 10 instruction means in response to a determination that each of said entities includes a native attribute, for removing said native attribute from each of said entities; and

instruction means for generating equivalent entities that include no native attributes.

15

30. The product according to claim 27, wherein said instruction means for identifying all public entities included in each of said public classes further comprises instruction means for identifying all public methods
20 included in each of said public classes.

31. The product according to claim 27, wherein said instruction means for identifying all public entities included in each of said public classes further comprises
25 instruction means for identifying all public parameters included in each of said public classes.

32. The product according to claim 27, wherein said instruction means for identifying all public entities
30 included in each of said public classes further comprises instruction means for identifying all public fields included in each of said public classes.

Docket No. AUS920010996US1

33. The product according to claim 27, wherein said instruction means for identifying all public classes included in said first package further comprises instruction means for identifying all public classes
5 included in a Java Archive file.

34. The product according to claim 27, wherein said instruction means for identifying all public classes included in said first package further comprises
10 instruction means for identifying all public classes included in said first package utilizing a java.util.jar utility.

35. The product according to claim 27, wherein said
15 instruction means for each of said public classes, for identifying all public entities included in each of said public classes further comprises instruction means for each of said public classes, for identifying all public entities included in each of said public classes
20 utilizing Java Reflection.

36. The product according to claim 27, wherein said instruction means for generating an equivalent public class for each of said identified public classes further
25 comprises instruction means for generating a separate .java file for each of said identified public classes.

37. The product according to claim 36, wherein said instruction means for compiling each of said equivalent
30 public classes further comprises instruction means for compiling each said .java file.

20200109 16:24:00

Docket No. AUS920010996US1

38. The product according to claim 37, wherein said instruction means for generating a compilation interface for said first package including each of said compiled equivalent public classes further comprises:

5 instruction means for generating a compilation Java Archive file; and

instruction means for storing each said compiled .java file in said compilation Java Archive file.

10 39. The product according to claim 27, further comprising instruction means for utilizing said compilation interface within an Integrated Development Environment.

204010996US1